Research & Review: Machine Learning and Cloud Computing
e-ISSN: 2583-4835
Volume-1, Issue-3 (September-December, 2022)
www.matjournals.com
https://doi.org/10.46610/RRMLCC.2022.v01i03.002

A Review on Real-Time Traffic Sign Recognition with Voice Warnings

Priya Surana[1], Prakash Sawant[2], Harsh Wangikar[3*], Napul Labde[4], Akshat Shah[5]
[1]Assistant Professor, Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India
[2,3,4,5]UG Student, Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India

[*]Corresponding Author: harsh.wangikar21@pccoepune.org

ABSTRACT
Road signs are essential for providing information to drivers. Understanding road signs are essential for ensuring traffic safety because doing so can stop 4484 accidents. The identification of traffic signs has been the focus of research in recent decades. Accurate real-time recognition is the cornerstone of a robust but underdeveloped traffic sign recognition system. This study provides drivers with real-time voice-advice traffic sign recognition technology. This system is composed of two subsystems. Using a trained convolutional neural network, the first recognizes and detects traffic signs (CNN). When the system notices a particular traffic sign, the text-to-speech engine is employed to play a voice message to the driver. An efficient- CNN model is built on the reference data set using deep learning methods for search and real-time search. This system's advantage is that it recognizes traffic signs and guides the car even if the driver overlooks, ignores, or doesn't understand them. Say. These technologies are also necessary for the development of autonomous vehicles.

Keywords- Convolutional neural networks, Pyttsx3, Region-based Convolutional Neural Network, Traffic sign detection and recognition, YOLO, YOLOv5

INTRODUCTION

Every country has implemented the required road laws and regulations to ensure safety as cars have become a necessary mode of transportation. One of them is a traffic sign, which informs motorists of the regulations that must be followed in that specific area. A traffic sign's objective is to quickly and accurately convey a message while needing little reading comprehension. Drivers' ignorance of traffic signs, which eventually causes traffic accidents, has been linked to several issues, including negligence, inattention, unfamiliarity, accidentally or purposely ignoring traffic signals, distracting driving behaviors, and others. Furthermore, because they are unfamiliar with the numerous road signs found in urban areas, drivers in rural areas may find it challenging to interpret the information included in a particular traffic sign. Certain traffic signs are frequently disregarded by drivers who think they are unnecessary. The drivers' various mindsets are another factor contributing to this ignorance. Traffic sign ignorance or unfamiliarity could cause serious accidents and possibly result in fatalities[1].

Because of the quicker detection rates and improved accuracy of the model, the system can be used to identify traffic signs in real-time. Narrating a traffic sign's message can be useful for drivers while they are on the road. The audio narration can overcome the issues of missing the traffic signs, not understanding them, and their complexity. To overcome the aforementioned concerns, this study suggests a method to accurately detect and recognize traffic signs in real-time while also narrating the signs to the drivers. Both autonomous and vehicle assistance systems can use this type of system. The system is built using the YOLO Convolutional Neural Network (CNN) model architecture.

YOLO is an open-source object identification method that utilizes the core modules of the Darknet and ImageNet neural networks. YOLO employs bounding boxes over image characteristics to combine many object detection components into a single neural network. YOLO uses convolutional neural networks (CNN) to swiftly maintain high detection accuracy. YOLO predicts the bounding boxes at each network based on the context of

https://doi.org/10.46610/RRMLCC.2022.v01i03.002

the items. The output, which is the single neural network, appears when these detections are defined as a regression problem in YOLO. Since 2016, YOLO has seen several developments. There are two options: the YOLOv2 or YOLO9000, and the YOLOv3.

YoloV5, which made it straightforward to capture diverse patterns with different resolutions by doing the convolutional procedure with heterogeneous kernel sizes. As a result, the traffic signs can still be seen despite their distorted shapes and colors at different viewing angles and distances[2].

RELATED WORK

Recently, the focus of study in the academic community has switched to autonomous and assisted driving. Traffic sign detection and recognition have been a key research field among them. Traffic sign recognition relies on three popular techniques: methods based on color, methods based on geometry, and methods based on machine learning. The HSV (Hue, Saturation, and Value) transformation is one of the most popular methods for color-based detection. Although color-based algorithms are computationally efficient, their accuracy diminishes due to bad lighting, shifting weather, lit environments, and backgrounds with similar colors. The shape-based detections extract shapes like triangles, rectangles, and circles where the traffic signs are projected to be using techniques such as the Hough transform and Edges with Harrlike features. This tactic fails when shapes that resemble traffic signs appear. As a result of the drawbacks of conventional procedures, the research community has now focused on deep learning techniques. Recently, Deep CNN has proven its capacity to deliver findings that are dependable, rapid, and accurate. Convolutional neural networks have the advantage over traditional techniques in that they can learn features from a large number of training cases without the requirement for preprocessing. High-end graphics processing units have aided in the creation of numerous convolutional neural network designs, such as R-CNN (Region-based Convolutional Neural Networks), YOLO (You Only Look Once), and SSD (Single Shot Multibox Detection) (GPU)[3].

The Belgian traffic sign dataset (Belgium TS), the German traffic sign detection benchmark (GTSDB), and the German traffic sign recognition dataset are now the most frequently used public traffic sign detection datasets (GTSRB). The 900 high-resolution, natural scene pictures in the GTSDB dataset contain traffic signs that are between 16 by 16 and 128 by 128 pixels in size. The pixel size for each image is 1360 by 800. After 600 photographs were used for training, 300 were utilized for testing. In the GTSRB dataset, 39209 photos are utilized for training and 12630 images are used for testing. German traffic signs are divided into three main categories and 43 subcategories, with sizes ranging from 15 by 15 to 250 by 250 pixels. The Belgium Traffic Signs dataset consists of two substantial datasets (BelgiumTS). There are two: a dataset for classification and another for detection (BTSD) (BTSC). The BTSD dataset consists of 5905 train images and 3101 test images. The 63 subclasses and three primary categories used to categorize the traffic signs in BelgiumTS are, along with the 4591 train photos and 2534 test images in the BTSC collection. These traffic sign statistics mostly apply to European traffic signs as shown in Fig.1.

Figure 1: *Traffic signs are taken into consideration.*

In addition to a voice assistance message to narrate the detected sign to the driver, this research proposes a novel one-stage CNN technique built on YOLO architecture. Although deep neural network methods for identifying and detecting traffic signs have been implemented[4].

METHODOLOGY

Convolutional Neural Networks

CNN is now widely recognized as the

https://doi.org/10.46610/RRMLCC.2022.v01i03.002

most cutting-edge form of deep learning applicable to the field of computer vision. A mathematical model that is modeled after the core components of neurons is referred to as a "neural network", and its name comes from the term. A tremendous number of artificial neurons are used in the construction of a deep neural network. These neurons are joined in a network and stacked in layers. It begins with vectors as its inputs, makes its way through the various levels of the network, and finally concludes what the output will be. A convolutional neural network, also known as a CNN, is a type of deep neural network that consists of three distinct layers: convolutional layers, pooling layers, and fully connected layers. The first two varieties of these are concerned with the process of obtaining characteristics, whereas the completely connected layers map the qualities that have been extracted into classes. Several distinct configurations of convolutional neural networks have been put into action in the course of the photo-detection and recognition process as shown in Fig. 2 [5].

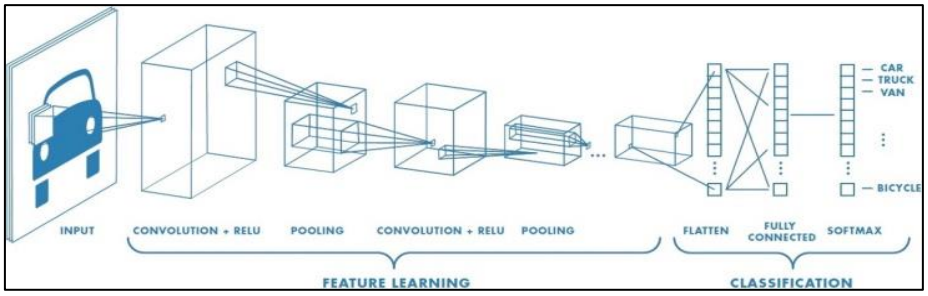

Figure 2: *CNN architecture.*

R-CNN

RCNN is an abbreviation for a methodology that consists of two levels. At the beginning of the procedure, a list of possible locations inside an image that could hold an item is compiled. This list is used later on in the process. Following the completion of the first step, the object is arranged into discrete zones inside each area in which it was discovered. The RCNN detector starts by initially producing region suggestions by using an approach that is comparable to Edge Boxes. This is where it gets its name. The image is cropped and enlarged so that the elements that were outlined in the previous step can be removed. Following the trimming, cropping, and shrinking of the segments, CNN moves on to the next step of classifying them. A support vector machine (SVM) that has been trained using CNN features are utilized to assist in the final step of the process, which involves refining the region proposal bounding boxes [6].

The train RCNN Object Detector function is utilized when going through the process of training the RCNN to identify objects. The result of calling this function will be an entity or entity that has the name RCNN Object Detector. This entity or entity can recognize as shown in Fig 3.

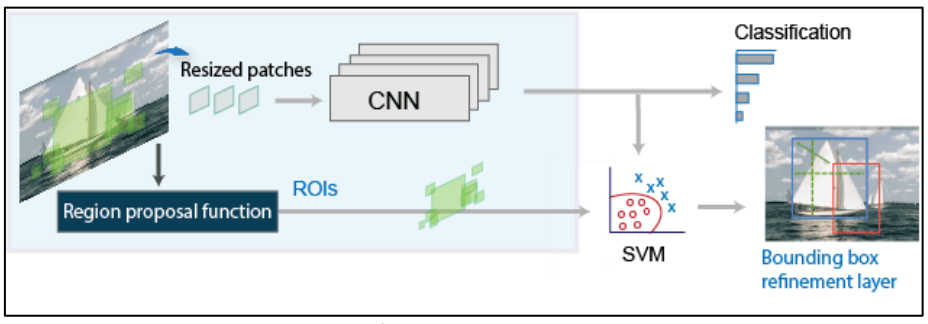

Figure 3: R-CNN

Fast R-CNN

The Fast RCNN detector, which operates in a manner analogous to that of Edge Boxes, also produces region recommendations. In contrast to the RCNN detector, the Fast

https://doi.org/10.46610/RRMLCC.2022.v01i03.002

RCNN detector processes the full image whilst the RCNN detector reduces and resizes the region proposals. Although an RCNN detector needs to classify each region, rapid RCNN does so by pooling CNN features relevant to each area proposal. Because it uses the same calculations for overlapping regions as the RCNN detector, the Fast RCNN detector is superior to it. If you want to train a Fast RCNN entity identifier, you should use the train Fast RCNN Object Detector function. The function produces a rapid RCNN Object Detector as its output, which is capable of identifying entities contained within a Fig.[7].

Faster R-CNN

As an alternative to employing approaches from the outside such as Edge Boxes, the Faster RCNN identifier integrates a region proposal network (RPN) to provide area recommendations immediately within the network. The RPN will search for the objects using the anchor boxes. Your data are provided with more specific and expedient area suggestions generated by the network. You can train a Faster RCNN object detector by utilizing the function known as train Faster RCNN Object Detector. The function's result is a more efficient RCNN Object Detector that can identify entities contained inside a Fig.4.

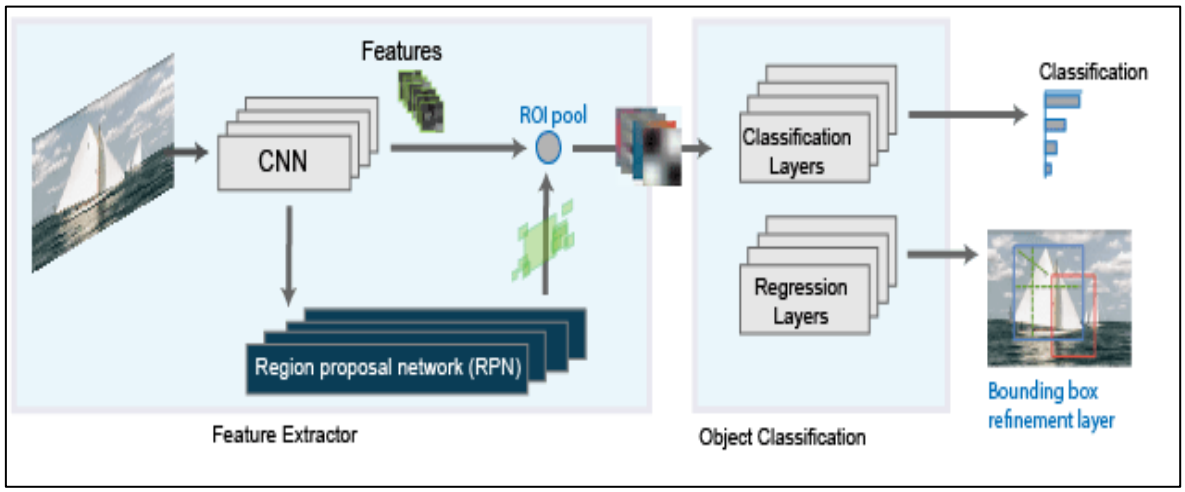

Figure 4: *Faster R-CNN*

YOLO

To produce the desired result—a single neural network—YOLO structures these detections as a regression problem. Since 2016, YOLO has seen a number of developments. There are two options: the YOLOv2 or YOLO9000, and the YOLOv3. In YOLO, an image is transmitted through a series of network layers, where predictions are generated about the bounding boxes containing objects at each layer. The output of the network is a single neural network. The total detection method consists of four basic phases. The algorithm initially divides the image into (n) S grids, and each grid is tasked with locating the object of interest based on a predicted confidence score[8]. An image is first divided into a (n) S S grid, similar to YOLO. The image is then resized to account for the discovery of a small object. After that, a bounding box is anticipated (known as an anchor box). In contrast, depending on the previous anchor box, YOLOv3 generates additional k bounding boxes for each grid. To improve the detection of each object in the bounding box, its regression approach now adds the loss function. The optimum bounding box is determined by the IOU values in a way similar to YOLO. YOLO is not as accurate as YOLOv3. This is because it has a distinct deep architecture dubbed Darknet-53, which contains 53 trained CNN layers instead of YOLOv2's 19 and 53 total layers. At the detection stage, these 53 layers are stacked on top of one another to produce a total of 106 layers in the convolutional architecture as shown in Fig. 5.

https://doi.org/10.46610/RRMLCC.2022.v01i03.002

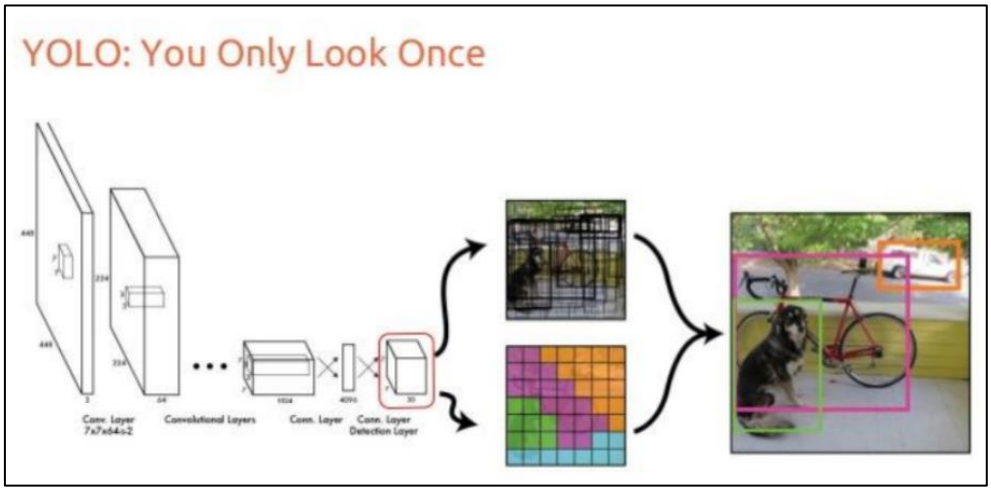

Figure 5: *YOLO architecture.*

Audio Narration

For the system to function as intended, an audio feedback system that may provide a narration of the observed traffic signs has been added to the system. Given that the detections are being carried out in real-time, the audio outputs ought to should likewise be transmitted in the same fashion. To achieve this goal, the detections and audio outputs are arranged to work in conjunction with one another rather than independently. Because of this, anytime an object is recognized, the appropriate verbal feedback for that sign is also provided at the same time. This ensures that the feedback is received promptly. The Python Text-to-Speech version 3 (pyttsx3) libraries are frequently utilized when it comes to the production of audio narrations. The user's chosen language can be quickly chosen from a drop-down menu of available alternatives. The algorithm that plays the particular voice is updated with the detected sign whenever there is a new detection made in the frame. When a new detection is made, this will always take place[9].

Pyttsx3 is the abbreviation for the Python text-to-speech package. It is compatible with Python versions 2 and 3, unlike other libraries, and it may be used without an internet connection. The process of converting written text into spoken word is known as "Text-to-speech" software. The text is first analyzed and processed using Natural Language Processing (NLP), and then using "Digital Signal Processing" (DSP) technology, the processed text is converted into a synthesized vocal representation of the written text. The text of a traffic sign was converted into a computer-generated voice using text-to-speech technology, which was used in this case in the form of a simple application that, when launched, reads the text to the useras shown in Fig. 6.

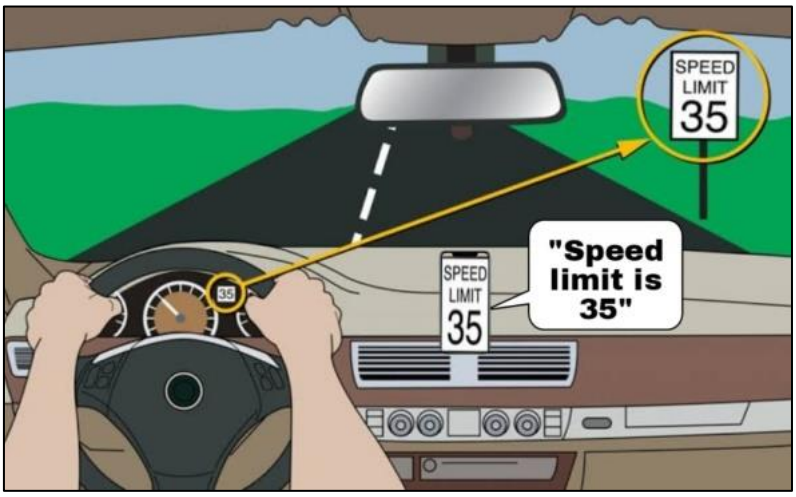

Figure 6: *Voice output demo.*

https://doi.org/10.46610/RRMLCC.2022.v01i03.002

Datasets

The system was initially developed using the German Traffic Sign Detection Benchmark (GTSDB) dataset to build a CNN model. The dataset contains 900 images in total, 600 of which are training images and 300 are validation images. There could be 0 to 6 different traffic sign types in one image. In this collection, traffic signs are shown from every angle imaginable and under a wide range of illumination conditions. Traffic sign occurrences have been divided into the following four groups: dangerous, prohibitory, mandatory, and others. The dataset comprises annotations that are in CSV format, and by using a Python-developed technique, they are converted to YOLO format. The labeling program was used to validate the annotations once they had been converted to YOLO format. The high-resolution images that were acquired from the Mapillary Traffic Sign Dataset, which comprises more than 100,000 unique shots, were used to validate the ensuing models. Nearly 300 different types of traffic signs may be found worldwide in practically every continent. The collection's photos were taken in a variety of weather conditions, including rain, sun, snow, dawn, midday, and night-time, among others. Since the dataset is labeled with 300 different classes, the number of images available for some classes was insufficient for training. The regulation type and the warning type of that particular sign are now grouped to reflect the same sign as a result of the classes being integrated with this fashion. The annotations had to be converted from the JSON file format into the YOLO file format; hence a different algorithm had to be devised to select the desired classes [10]. The labelling tool was used to assess whether the conversion was successful or not as shown in Fig. 7 ,8 and 9.

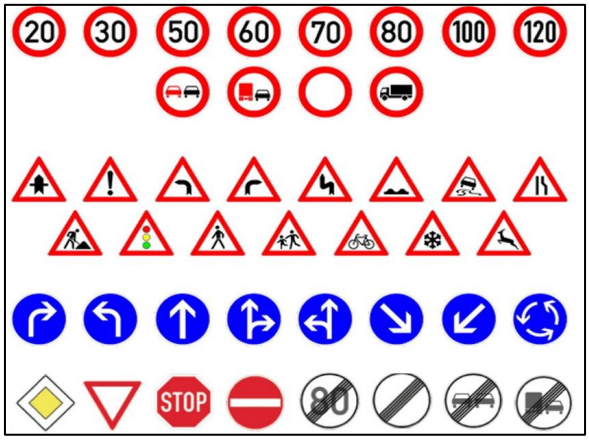

Figure 7: *Datasets images* (Lihua, 2017).

COMPARATIVE STUDY

Algorithm	Features	Prediction time / image	Limitations
CNN	Divides the image into multiple regions and then classifies each region into various classes.	–	Needs a lot of regions to predict accurately and hence high computation time.
R-CNN	Uses selective search to generate regions. Extracts around 2000 regions from each image.	40-50 seconds	High computation time as each region is passed to the CNN separately. Also, it uses three different models for making predictions.
Fast R-CNN	Each image is passed only once to the CNN and feature maps are extracted. Selective search is used on these maps to generate predictions. Combines all the three models used in R-CNN together.	2 seconds	Selective search is slow and hence computation time is still high.
Faster R-CNN	Replaces the selective search method with region proposal network (RPN) which makes the algorithm much faster.	0.2 seconds	Object proposal takes time and as there are different systems working one after the other, the performance of systems depends on how the previous system has performed.

Figure 8: *Comparison between various CNN-based algorithms* (Sharma, 2018).

https://doi.org/10.46610/RRMLCC.2022.v01i03.002

	YOLOv3	YOLOv4	YOLOv5
Neural Network Type	Fully convolution	Fully convolution	Fully convolution
Backbone Feature Extractor	Darknet-53	CSPDarknet53	CSPDarknet53
Loss Function	Binary cross entropy	Binary cross entropy	Binary cross entropy and Logits loss function
Neck	FPN	SSP and PANet	PANet
Head	YOLO layer	YOLO layer	YOLO layer

Figure 9: *Comparison between various YOLO versions* (Nepal, 2022).

CONCLUSION

Raising traffic and driver safety is especially facilitated by the driving assistance strategy of traffic sign recognition. Using a range of techniques and methods, such as binarization, region of interest (ROI), and pixel classification, it has been effectively utilized to recognize traffic signs. Many algorithms have been used, including CNN, RCNN, YOLO, YOLO5, etc. Finding a traffic sign may provide some difficulties because object recognition in outdoor environments is dependent on factors like illumination and weather. Moving objects present a detecting challenge, including moving store signage, moving pedestrians, moving bicycles, and moving cars. Traffic signs' colors deteriorate over time as a result of solar aging and paint reaction. The quality of pictures shot from moving cars is lower due to motion blur and vehicle vibration. These elements make it challenging to recognize traffic signs in such situations with accuracy.

Several algorithms, including CNN, YOLO, and Faster RCNN, are contrasted. We found that the YOLOv3 and YOLOv5 algorithm is capable of delivering more accurate findings and giving more information about the image as a result of the comparison investigation.

REFERENCES

1. D Mijić, M Brisinello, M Vranješ and R Grbić (2020). Traffic sign detection using YOLOv3. *2020 IEEE 10th International Conference on Consumer Electronics (ICCE-Berlin)*. IEEE, Available at: https://doi.org/10.1109/ICCE-Berlin50680.2020.9352180.
2. W-Noorshahida Mohd-Isa, Md-Shakif Abdullah, M Sarzil, et al (2020). Detection of Malaysian traffic signs via modified YOLOv3 algorithm. *2020 International Conference on Data Analytics for Business and Industry: Way towards a Sustainable Economy (ICDABI)*. IEEE, Available at: https://doi.org/10.1109/ICDABI51230.2020.9325690.
3. Y Sun, P Ge and D Liu (2019). Traffic sign detection and recognition based on convolutional neural network. *2019 Chinese Automation Congress (CAC)*. IEEE, Available at: https://doi.org/10.1109/CAC48633.2019.8997240.
4. D Karthikeyan, C Enitha, S Bharathi and K Durkadevi (2020). Traffic sign detection and recognition using image processing, *International Journal of Engineering Research & Technology*, 8(8), 1-4, Available at: https://www.ijert.org/research/traffic-sign-detection-and-recognition-using-image-processing-IJERTCONV8IS08019.pdf.
5. M Manawadu and U Wijenayake (2021). Voice-assisted real-time traffic sign recognition system using convolutional neural network. *2021 International Conference on Advanced Research in Computing*. ICARC, Available at: https://www.researchgate.net/publication/353609044_Voice-Assisted_Real-Time_Traffic_Sign_Recognition_System_Using_Convolutional_Neural_Network.
6. S Nanda, P More, M Shimpi, et al (2019). Traffic sign recognition system: A survey, *International Research Journal of Engineering and Technology*, 6(3), 7291-7294, Available at: https://www.irjet.net/archives/V6/i3/IRJET-V6I3829.pdf.
7. D Kothadiya, N Pise and M Bedekar (2020). Different methods review for

https://doi.org/10.46610/RRMLCC.2022.v01i03.002

speech to text and text to speech conversion, *International Journal of Computer Applications*, 175(20), 9-12, Available at: https://www.ijcaonline.org/archives/volume175/number20/kothadiya-2020-ijca-920727.pdf.

8. Sharma, P. (2018, November 4). Retrieved from analytics vidhya: https://medium.com/analytics-vidhya
9. Nepal, U. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors*.
10. Lihua, W. (2017). *Traffic sign recognition and classification with modified residual networks*. 10.1109/SII.2017.8279326.